The Design of Collectives of Agents to Control Non-Markovian Systems

John W. Lawson and David H. Wolpert
NASA Ames Research Center
MS 269-2
Moffett Field, CA 94035
{lawson,dhw}@ptolemy.arc.nasa.gov

Abstract

The "Collective Intelligence" (COIN) framework concerns the design of collectives of reinforcement-learning agents such that their interaction causes a provided "world" utility function concerning the entire collective to be maximized. Previously, we applied that framework to scenarios involving Markovian dynamics where no re-evolution of the system from counter-factual initial conditions (an often expensive calculation) is permitted. This approach sets the individual utility function of each agent to be both aligned with the world utility, and at the same time, easy for the associated agents to optimize. Here we extend that approach to systems involving non-Markovian dynamics. In computer simulations, we compare our techniques with each other and with conventional "team-games". We show whereas in team games performance often degrades badly with time, it steadily improves when our techniques are used. We also investigate situations where the system's dimensionality is effectively reduced. We show that this leads to difficulties in the agents' ability to learn. The implication is that "learning" is a property only of high-enough dimensional systems.

Introduction

In this paper we are concerned with large distributed collectives of interacting goal-driven computational processes, where there is a provided 'world utility' function that rates the possible behaviors of that collective (Wolpert, Tumer, & Frank 1999; Wolpert & Tumer 1999). We are particularly concerned with such collectives where the individual computational processes use machine learning techniques (e.g., Reinforcement Learning (RL) (Kaelbing, Littman, & Moore 1996; Sutton & Barto 1998; Sutton 1988; Watkins & Dayan 1992)) to try to achieve their individual goals. We represent those goals of the individual processes as maximizing an associated 'payoff' utility function, one that in general can differ from the world utility.

In such a system, we are confronted with the following inverse problem: *How should one initialize/update the payoff utility functions of the individual processes so that the ensuing behavior of the entire collective achieves large values of the provided world utility?* In particular, since in truly large systems detailed modeling of the system is usually impossible, how can we avoid such modeling? Can we instead leverage the simple assumption that our learnering algorithms are individually fairly good at what they do to achieve a large world utility value?

We are concerned with payoff utility functions that are "aligned" with the world utility, in that modifications a player might make that would improve its payoff utility also must improve world utility.[1] Fortunately the equivalence class of such payoff utilities extends well beyond team-game utilities. In particular, in previous work we used the COllective INtelligence (COIN) framework to derive the 'Wonderful Life Utility' (WLU) payoff function (Wolpert & Tumer 1999) as an alternative to a team-game payoff utility. The WLU is aligned with world utility, as desired. In addition though, WLU overcomes much of the signal-to-noise problem of team game utilities (Tumer & Wolpert 2000; Wolpert, Tumer, & Frank 1999; Wolpert & Tumer 1999; Wolpert, Wheeler, & Tumer 2000).

In a recent paper, we extended the COIN framework with an approach based on Transforming Arguments Utility functions (TAU) before the evaluation of those functions (Wolpert & Lawson 2002). The TAU process was originally designed to be applied to the individual utility functions of the agents in systems in which the world utility depends on the final state in an episode of variables outside the collective that undergo Markovian dynamics, with the update rule of those variables reflecting the state of the agents at the beginning of the episode. This is a very common scenario, obtaining whenever the agents in the collective act as control signals perturbing the evolution of a Markovian system.

In the pre-TAU version of the COIN framework, to achieve good signal-to-noise for such scenarios requires knowing the evolution operator. However it also might require re-evolving the system from counter-factual initial states of the agents to evaluate each agent's reward for a particular episode. This can be computationally expensive. With TAU utility functions no such re-evolving is needed; the observed history of the system in the episode is transformed in a relatively cheap calculation, and then the util-

[1] Such alignment can be viewed as an extension of the concept of incentive compatibility in mechanism design (Fudenberg & Tirole 1991) to non-human agents, off-equilibrium behavior, etc.

ity function is evaluated with that transformed history rather than the actual one.

The TAU process has other advantages that apply even in scenarios not involving Markovian dynamics. In particular it allows us to employ the COIN framework even when not all arguments of the original utility function are observable, due for example to communication limitations. In addition, certain types of TAU transformations result in utility functions that are not exactly aligned with the world utility, but have so much better signal-to-noise that the collective performs better when agents use those transformed utility functions than it does with exactly aligned utility functions.

Here we investigate the extension of the TAU process to systems with non-Markovian dynamics where the world utility is the same function of the state of the system at every moment in time. To do this we have the agents operate on very fast time-scales compared to that dynamics, i.e., have the time-steps at which they make their successive moves be very closely packed. We also have the moves of the agents consist of very small perturbations to the underlying variables of the system rather than the direct setting of those variables. Now since the world utility is defined for every moment in time, there is a surface taking the values of those underlying variables at any time-step to the associated value of the world utility. So the problem for the agents is one of traversing that surface to try to get to values of the underlying variables to have a good associated world utility.

Since the time-scales are so small though, we can approximate the effects of the agents' moves at any time-step of the value of the world utility at the next time-step as though the intervening evolution were linear (Markovian). Now, as in the original TAU work, assume for simplicity that that linear dynamics is known for each such time-step. Then at each time-step the problem is reduced to the exact same one that was addressed in that original TAU work.

Unlike in that original work though, here the linear relation between the moves of the agents and the resultant value of the world utility at the next time-step changes from one time-step to the next, as both the underlying variables of the system change as does the associated gradient. Accordingly, the mapping the agents are trying to learn from their moves to the resultant rewards changes in time.

Here we do not confront this nonstationarity. We use a set of computer experiments to compare use of the TAU process to set the utility functions of agents to the alternative conventional approach of "team games" in this non-Markovian domain. We verify that the TAU process outperforms this alternative. In particular, in many experiments the team game resulted in world utility values that *decrease* with time, i.e., the agents steer the underlying variables to worse and worse values. In contrast, the TAU process steer the underlying variables in such a way that improved world utility with time.

We also investigate what happens as the underlying system is modified so that the moves of the individual agents become less and less consequential to the dynamics. Intuitively, one would expect in such a case that the system's effective dimensionality gets reduced, while the agents also have a harder time learning. We present tentative evidence corroborating this prediction. The implication is that "learning" is a property only of high-enough dimensional systems.

The Mathematics of Collective Intelligence

We view the individual agents in the collective as players involved in a repeated game.[2] Let Z with elements ζ be the space of possible joint moves of all players in the collective in some stage. We wish to search for the ζ that maximizes a provided **world utility** $G(\zeta)$. In addition to G we are concerned with utility functions $\{g_\eta\}$, one such function for each variable/player η. We use the notation $\hat{\eta}$ to refer to all players other than η.

Intelligence and the central equation

We wish to "standardize" utility functions so that the numeric value they assign to a ζ only reflects their ranking of ζ relative to certain other elements of Z. We call such a standardization of an arbitrary utility U for player η the "**intelligence** for η at ζ with respect to U". Here we will use intelligences that are equivalent to percentiles:

$$\epsilon_U(\zeta : \eta) \equiv \int d\mu_{\zeta_\eta}(\zeta')\Theta[U(\zeta) - U(\zeta')] \,, \qquad (1)$$

where the Heaviside function Θ is defined to equal 1 when its argument is greater than or equal to 0, and to equal 0 otherwise, and where the subscript on the (normalized) measure $d\mu$ indicates it is restricted to ζ' sharing the same non-η components as ζ. In general, the measure must reflect the type of system at hand, e.g., whether Z is countable or not, and if not, what coordinate system is being used. Other than that, any convenient choice of measure may be used and the theorems will still hold. Intelligence value are always between 0 and 1.

Our uncertainty concerning the behavior of the system is reflected in a probability distribution over Z. Our ability to control the system consists of setting the value of some characteristic of the collective, e.g., setting the functions of the players. Indicating that value by s, our analysis revolves around the following central equation for $P(G \mid s)$, which follows from Bayes' theorem:

$$P(G \mid s) = \int d\vec{\epsilon}_G P(G \mid \vec{\epsilon}_G, s)$$
$$\int d\vec{\epsilon}_g P(\vec{\epsilon}_G \mid \vec{\epsilon}_g, s) P(\vec{\epsilon}_g \mid s) \,, \qquad (2)$$

where $\vec{\epsilon}_g \equiv (\epsilon_{g_{\eta_1}}(\zeta : \eta_1), \epsilon_{g_{\eta_2}}(\zeta : \eta_2), \cdots)$ is the vector of the intelligences of the players with respect to their associated functions, and $\vec{\epsilon}_G \equiv (\epsilon_G(\zeta : \eta_1), \epsilon_G(\zeta : \eta_2), \cdots)$ is the vector of the intelligences of the players with respect to G.

Note that $\epsilon_{g_\eta}(\zeta : \eta) = 1$ means that player η is fully rational at ζ, in that its move maximizes its utility, given the moves of the players. In other words, a point ζ where

[2]The full mathematics of the COIN framework, however, extends significantly beyond what is needed to address such games. See (Wolpert & Tumer 2001).

$\epsilon_{g_\eta}(\zeta : \eta) = 1$ for all players η is one that meets the definition of a game-theory Nash equilibrium (Fudenberg & Tirole 1991). Note that consideration of points ζ at which not all intelligences equal 1 provides the basis for a model-independent formalization of bounded rationality game theory, a formalization that contains variants of many of the theorems of conventional full-rationality game theory (Wolpert 2001a). On the other hand, a ζ at which all components of $\vec{\epsilon}_G = 1$ is a local maximum of G (or more precisely, a critical point of the $G(\zeta)$ surface).

If we can choose s so that the third conditional probability in the integrand is peaked around vectors $\vec{\epsilon}_g$ all of whose components are close to 1, then we have likely induced large intelligences. If in addition the second term is peaked about $\vec{\epsilon}_G$ equal to $\vec{\epsilon}_g$, then $\vec{\epsilon}_G$ will also be large. Finally, if the first term is peaked about high G when $\vec{\epsilon}_G$ is large, then our choice of s will likely result in high G, as desired.

Intuitively, the requirement that the utility functions have high "signal-to-noise" (an issue not considered in conventional work in mechanism design) arises in the third term. It is in the second term that the requirement that the utility functions be "aligned with G" arises. In this work we concentrate on these two terms, and show how to simultaneously set them to have the desired form.

Details of the stochastic environment in which the collective operates, together with details of the learning algorithms of the players, are reflected in the distribution $P(\zeta)$ which underlies the distributions appearing in Equation 2. Note though that *independent of these considerations*, our desired form for the second term in Equation 2 is assured if we have chosen utility utilities such that $\vec{\epsilon}_g$ equals $\vec{\epsilon}_G$ exactly *for all* ζ. We call such a system *factored*. In game-theory language, the Nash equilibria of a factored collective are local maxima of G. In addition to this desirable equilibrium behavior, factored collectives automatically provide appropriate off-equilibrium incentives to the players (an issue rarely considered in game theory / mechanism design).

Opacity

We now focus on algorithms based on utility functions $\{g_\eta\}$ that optimize the signal/noise ratio reflected in the third term, subject to the requirement that the system be factored. To understand how these algorithms work, given a measure $d\mu(\zeta_\eta)$, define the **opacity** at ζ of utility U as:

$$\Omega_U(\zeta : \eta, s) \equiv \int d\zeta' J(\zeta' \mid \zeta) \frac{|U(\zeta) - U(\zeta'_{\neg\eta}, \zeta_\eta)|}{|U(\zeta) - U(\zeta_{\neg\eta}, \zeta'_\eta)|}, \quad (3)$$

where J is defined in terms of the underlying probability distributions,[3] and $(\zeta'_\eta, \zeta_\eta)$ is defined as the worldline whose η components are the same as those of ζ' while its η components are the same as those of ζ ((Wolpert & Tumer 2001)).

The denominator absolute value in the integrand in Equation 3 reflects how sensitive $U(\zeta)$ is to changing ζ_η. In contrast, the numerator absolute value reflects how sensitive $U(\zeta)$ is to changing $\zeta_{\neg\eta}$. So the smaller the opacity of a utility function g_η, the more $g_\eta(\zeta)$ depends only on the move of player η, i.e., the better the associated signal-to-noise ratio for η. Intuitively then, lower opacity should mean it is easier for η to achieve a large value of its intelligence.

To formally establish this, we use the same measure $d\mu$ to define opacity as the one that defined intelligence. Under this choice expected opacity bounds how close to 1 expected intelligence can be (Wolpert & Tumer 2001):

$$E(\epsilon_U(\zeta : \eta) \mid s) \leq 1 - K, \text{where}$$
$$K \leq E(\Omega_U(\zeta : \eta, s) \mid s). \quad (5)$$

So low expected opacity of utility g_η ensure that a necessary condition is met for the third term in Equation 2 to have the desired form for player η. While low opacity is not, formally speaking, also sufficient for $E(\epsilon_U(\zeta : \eta) \mid s)$ to be close to 1, in practice the bounds in Equation 5 are usually tight.

Difference Utilities

It is possible to solve for the set of all utilities that are factored with respect to a particular world utility. Unfortunately, in general it is not possible for a collective both to be factored and to have zero opacity for all of its players. However consider **difference** utilities, which are of the form

$$U(\zeta) = G(\zeta) - \Gamma(f(\zeta)) \quad (6)$$

where $\Gamma(f)$ is independent of ζ_η. Any difference utility is factored (Wolpert 2001b), and under benign approximations, $E(\Omega_u \mid s)$ is minimized over the set of such utilities by choosing

$$\Gamma(f(\zeta)) = E(G \mid \zeta_{\neg\eta}, s), \quad (7)$$

up to an overall additive constant. We call the resultant difference utility the **Aristocrat** utility (AU), loosely reflecting the fact that it measures the difference between a player's actual action and the average action.

The COIN Framework for Systems with Markovian Evolution

We consider games which consist of multi-step "episodes". Within each episode the entire system evolves in a Markovian manner from the initial moves of the players. We are interested in such games where some of the players η are not agents whose initial state is under control of a learning algorithm that we control, but rather constitute an "environment" for those controllable agents (i.e., where some of the players correspond to the state of nature).

Let A be the Markovian single step evolution operator of the entire system through an episode,

$$\vec{\zeta}_t = A\vec{\zeta}_{t-1} \quad (8)$$

Each component ζ^η_t, for example, could be a one-dimensional real number. The row vector A^η would then

[3] Writing it out in full, $J(\zeta' \mid \zeta) \equiv J(\zeta_\eta, \zeta' \mid \zeta_{\neg\eta}, s)/P(\zeta_\eta \mid \zeta_{\neg\eta}, s)$, with:

$$J(\zeta_\eta, \zeta' \mid \zeta_{\neg\eta}, s) \equiv \frac{P(\zeta_\eta \mid \zeta_{\neg\eta}, s)P(\zeta'_\eta \mid \zeta_\eta, s)\mu(\zeta'_\eta)}{2} + \frac{P(\zeta'_\eta \mid \zeta'_{\neg\eta}, s)P(\zeta_\eta \mid \zeta'_\eta, s)\mu(\zeta_\eta)}{2}. \quad (4)$$

be η's update rule. Alternatively, each agent could be represented by one of N symbolic values. In that case, $\vec{\zeta}_t$ would be given in a unary representation as a vector in $\mathcal{R}^{\mathcal{N}^{|\eta|}}$ (i.e. a Haar basis). Considering such large spaces are necessary to describe arbitrary, nonlinear dynamics as Markovian evolution. Here we will concentrate on the former case, where the moves of the players are all real numbers.

The full multiple time step evolution of an episode is given by single step operator in the usual way: Let

$$C = \begin{bmatrix} A \\ A^2 \\ A^3 \\ . \\ . \\ A^T \end{bmatrix}$$

where T is the number of time steps per episode. This operator applied to our initial state $\vec{\zeta}_0$ yields the entire "worldline" $\vec{\zeta}$, or time history, of the system.

$$\vec{\zeta} = C\vec{\zeta}_0. \tag{9}$$

We consider difference utility functions of the form

$$g_\eta(\vec{\zeta}) = G(C\vec{\zeta}_0) - \Gamma_\eta(F_\eta C\vec{\zeta}_0) \tag{10}$$

where G is the world utility function to be optimized. We will choose F_η so that the product $F_\eta C\vec{\zeta}_0$ is independent of agent η's actions. This is a necessary and sufficient condition for the associated difference utility $g_\eta(\vec{\zeta})$ to be factored with respect to the world utility G for any and all choices of Γ_η. In general, Γ_η can be chosen in such a way to optimize learnability. Here though, for simplicity, we choose $\Gamma_\eta = G$. Accordingly, application of the F_η operator is an instance of transforming the argument of the (second term of the) utility functions of the agents, i.e., it is a TAU process.

Observability restrictions

In practice, the full worldline of the system may not be fully observable to each agent. Such limited observability of a particular component may be determined by the problem. In other cases, due to communication constraints each agent is only allowed to observe a certain number of components, and must select which such components to observe, for example to optimize some auxiliary quantity like opacity. Similarly, the dynamics may not be known exactly to the agent; some rows of C may be uncertain to an agent, or simply cannot be determined. In these kinds of situations the g_η described above cannot be evaluated at the end of an episode by agent η, even if the value $G(\vec{\zeta}_t$ is globally broadcast to all agents.

The TAU approach outlined above is well-suited to address such situations. Formally, a decimated identity operator L can be defined whose diagonal elements are $\{0, 1\}$ depending on whether or not they are observable. The corresponding factored utility for agent η is

$$g_\eta(\vec{\zeta}_t) = G(\vec{\zeta}_t) - G(LF_\eta\vec{\zeta}_t), \tag{11}$$

where in general L may vary with η. Given global broadcast to all agents of the value of $G(\vec{\zeta}_t)$, for each agent to evaluate this type of g_η only requires that those components of $F_\eta\vec{\zeta}_t$ that are non-zero (and therefore can vary) after application of the L operator be observed.

This difference utility has two main sources of noise, one from potentially poor choice of the clamping operator, and the other from the use of L in the second (subtracted) term but not in the first. To address that latter source of noise we can impose limited observability on the first term in addition to the second one, getting

$$g_\eta(\vec{\zeta}_t) = G(L\vec{\zeta}_t) - G(LF_\eta\vec{\zeta}_t). \tag{12}$$

The new utility is not factored with respect to G. According to the central equation however, it may still result in better performance than when we don't have L in the first term, if the improvement in opacity more than offsets the loss of exact factoredness. In addition to the potential for such far superior opacity, this utility has the added advantage that now we don't even need to rely on global broadcast of $G(L\vec{\zeta}_t)$ to evaluate g_η.

The non-Markovian case

To address the general nonlinear problem, we assign each agent a real-valued number r_η. The state of the system $\vec{\zeta}_t$ is the Cartesian product of each agent's action and $\vec{r}_t$. Each agent can choose among three actions which add one of the values $\{\pm\Delta, 0\}$ to r_η. Nonlinear evolution then occurs to $\vec{r}$, to produce the value at the end of this episode, $\vec{\zeta}_{t+1} = \vec{c}_t(\vec{\zeta}_t)$. That value then serves as the argument of G.

Construction of factored utilities

$$g_\eta(\vec{\zeta}_{t+1}) = G(\vec{c}_t(\vec{\zeta}_t)) - G(\vec{c}_t(\hat{C}L\vec{\zeta}_t)) \tag{13}$$

requires that $\vec{c}_t(\vec{\zeta}_t)$ be independent of η's choice of action. One way to accomplish this to clamp (apply $\hat{C}L$) to $\vec{\zeta}_t$ and re-evolve the system. To avoid re-evolving the system, we approximate $\vec{c}_t(\hat{C}L\vec{\zeta}_t)$ with a Taylor Series expansion about the unclamped $\vec{\zeta}_t$ starting state:

$$\vec{c}_t(\hat{C}L\vec{\zeta}_t) = \vec{c}_t(\vec{\zeta}_t) + \Delta(\vec{\zeta}_t - \hat{C}L\vec{\zeta}_t) \cdot \vec{\nabla}_t\vec{c}_t(\vec{\zeta}_t). \tag{14}$$

Assuming not all components of $\vec{\zeta}_t$ equal 0, we can recast this as as the multiplication of a matrix times $\vec{\zeta}_t$, where that matrix is indexed by time. In doing this we reduce the system to the linear case, only with a time-dependent update matrix.

Note that varying Δ provides us a small parameter to control the expansion. It should also be noted that while this method requires that $\vec{c}_t(\vec{\zeta})$ be differentiable, the world utility G need not be.

Experiments

Numerical simulations were performed with 50 agents. After an initial 100-episode training period, agents selected initial actions in each subsequent episode with the same reinforcement learning algorithm used in our previous work.

All players experienced a quadratic/nonlinear update rule $\vec{c}(\vec{\zeta_0}) = \sum_{i,j} a_{i,j} r_0^i r_0^j$ that depends are agents "position" $\{r^i\}$. The coefficients are randomly generated. The world utility function was a spin glass,

$$G_T = \sum_{i<j} J_{ij} \zeta_T^i \zeta_T^j. \quad (15)$$

The agents are given a random initial starting point with $-1 < r_\eta < 1$. Because $\vec{c}$ is quadratic, $G(\vec{\zeta_t})$ is a quartic polynomial in N dimensions. Since the coefficients $\{a_{i,j}\}$ have random signs, the function G has as many increasing directions as it decreasing directions. The goal of the system is to traverse this high dimensional surface, find an increasing direction, and then follow that direction out to infinity.

We collected statistics by averaging runs over many randomly set coefficients $a_{i,j}$ and coupling constants J_{ij}. These runs were for systems whose first 25% and 75% components at the end of the episode are observable, given some canonical ordering of agents. We examined (Figure 1) world utility value vs. episode number for six utility functions:

1) TAU g for a fully observable system;

2) TAU g for 75 % observability, $g^{75\%}$;

3) The modification $g_{nf}^{75\%}$ giving a non-factored system, again with 75 % observability;

4) $g^{25\%}$ for a factored system with 25 % observability;

5) $g_{nf}^{25\%}$ for a non-factored system with 25 % observability;

6) The team game, where every $g_\eta = G$.

Even the results for limited observability clearly outperform the corresponding team game in which there is full observability. Furthermore, for 75% observability, the non-factored utilities (L in both terms) consistently outperform their factored counterpart. In these runs factoredness fell to approximately 90%. The improvement in performance due to better signal-to-noise more than outweighs the degradation due to loss in factoredness.

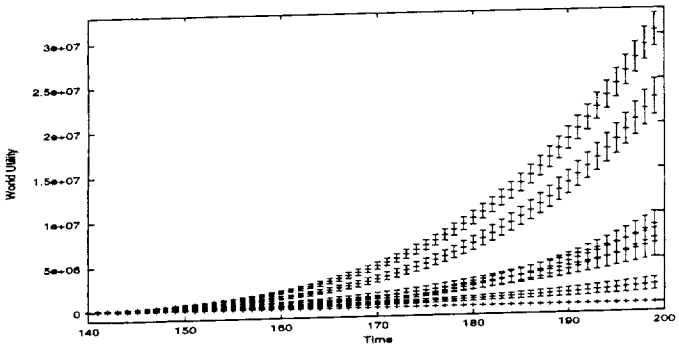

Figure 1: System performance for $N = 50$ agents using the Taylor Series method. The dynamics is governed by a quadratic function of the agents' "positions". The world utility G is a quartic in N dimensions. (upper two graphs are g and $g_{nf}^{75\%}$; middle two are $g_{nf}^{25\%}$ and $g^{75\%}$; lower two are $g^{25\%}$ and a team game G.) The initial training period is not shown.

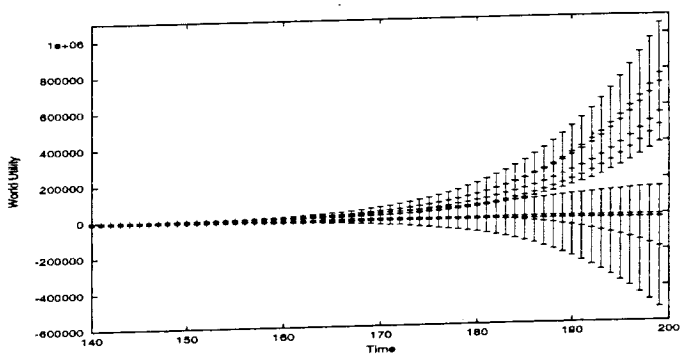

Figure 2: Taylor Series method where the quadratic coefficients have more $-$ than $+$ signs. (graphs: upper pair are g and $g_{nf}^{75\%}$; middle three are $g^{75\%}$, $g^{25\%}$, and the team game; lower is $g_{nf}^{25\%}$.) In this case, three of the limited observability utilities and the team game perform worse over time (i.e. their world utilities decrease).

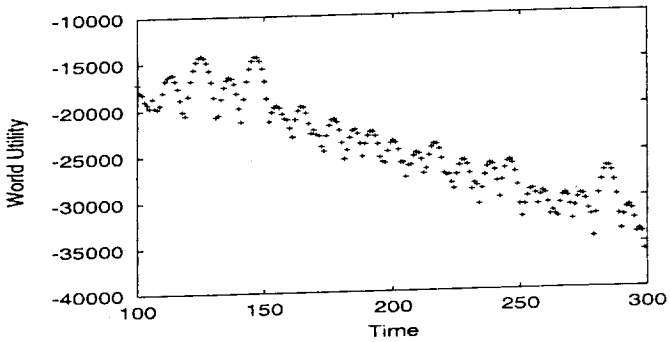

Figure 3:

It is interesting to adjust the ratio of $\pm$ signs in the coefficients of the polynomials. If we introduce, for example, more negative coefficients than positive, we expect the surface to preferentially turn down. The task for the agents becomes more challenging. We find (Figure 2), in fact, that three of the limited observability utilities perform worse over time (i.e. their world utility decreases). The team game also performs worse over time. In fact, not only does the team game give poor performance, but it fails altogether. The lowest noise TAU utilities g and $g_{nf}^{75\%}$ still give robust performance.

In this case, the team game gives worse performance than a random walk i.e. no learning is happening. In fact, the system executes essentially determistic, nonlinear behavior (Figure 3). Remarkably, as we increase the data aging parmeter (weighting more heavily data that appeared further in the past), the system becomes even more exotic, closely resembling a low-dimensional nonlinear system. By aging the data more severely, we effectively damp out a large portion of the degrees of freedom stored in the agents' training sets, hence the lower dimensionality. Learning, it would seem, is possible only in higher-dimensional systems.

Conclusion

We present a detailed extension of the $COIN$ framework to systems that undergo non-Markovian evolution. This builds on previous work where the Markovian case (Wolpert & Lawson 2002) was considered. The approach is applied to systems with nonlinear update rules using a perturbative technique. Results from numerical simulations find consistent, robust improvement of performance as compared to the conventional team game.

This framework naturally includes the case of limited observability. We found that even COIN-based utility functions constrained by limited observability often outperformed team game utilities having full observability. We also found a new class of nonfactored utilities that consistently outperformed their factored counterpart, due to improved signal-to-noise characteristics.

We find that the system's performance can depend on the characteristics of the surface being optimized. We show that in some situations a team game will fail altogether (i.e. its performance will degrade over time) while the corresponding TAU utility continues to perform well. In this "non-learning regime", the system executes interesting deterministic, nonlinear behavior, indicative of low-dimensional systems.

References

Boutilier, C.; Shoham, Y.; and Wellman, M. P. 1997. Editorial: Economic principles of multi-agent systems. *Artificial Intelligence Journal* 94:1–6.

Bradshaw, J. M., ed. 1997. *Software Agents*. MIT Press.

Caldarelli, G.; Marsili, M.; and Zhang, Y. C. 1997. A prototype model of stock exchange. *Europhysics Letters* 40:479–484.

Fudenberg, D., and Tirole, J. 1991. *Game Theory*. Cambridge, MA: MIT Press.

Hu, J., and Wellman, M. P. 1998. Multiagent reinforcement learning: Theoretical framework and an algorithm. In *Proceedings of the Fifteenth International Conference on Machine Learning*, 242–250.

Huberman, B. A., and Hogg, T. 1988. The behavior of computational ecologies. In *The Ecology of Computation*. North-Holland. 77–115.

Jennings, N. R.; Sycara, K.; and Wooldridge, M. 1998. A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems* 1:7–38.

Johnson, N. F.; Jarvis, S.; Jonson, R.; Cheung, P.; Kwong, Y. R.; and Hui, P. M. 1998. Volatility and agent adaptability in a self-organizing market. preprint cond-mat/9802177.

Kaelbing, L. P.; Littman, M. L.; and Moore, A. W. 1996. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research* 4:237–285.

Sandholm, T.; Larson, K.; Anderson, M.; Shehory, O.; and Tohme, F. 1998. Anytime coalition structure generation with worst case guarantees. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 46–53.

Sen, S. 1997. *Multi-Agent Learning: Papers from the 1997 AAAI Workshop (Technical Report WS-97-03*. Menlo Park, CA: AAAI Press.

Sutton, R. S., and Barto, A. G. 1998. *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Sutton, R. S. 1988. Learning to predict by the methods of temporal differences. *Machine Learning* 3:9–44.

Sycara, K. 1998. Multiagent systems. *AI Magazine* 19(2):79–92.

Tumer, K., and Wolpert, D. H. 2000. Collective intelligence and Braess' paradox. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 104–109.

Watkins, C., and Dayan, P. 1992. Q-learning. *Machine Learning* 8(3/4):279–292.

Wellman, M. P. 1993. A market-oriented programming environment and its application to distributed multicommodity flow problems. In *Journal of Artificial Intelligence Research*.

Wolpert, D., and Lawson, J. 2002. Designing agent collectives for systems with markovian dynamics. In *Proceedings of Autonomous Agents and MultiAgent Systems (AAMAS 2002)*. In press.

Wolpert, D. H., and Tumer, K. 1999. An Introduction to Collective Intelligence. Technical Report NASA-ARC-IC-99-63, NASA Ames Research Center. URL:`http://ic.arc.nasa.gov/ic/projects/coin_pubs.html`. To appear in Handbook of Agent Technology, Ed. J. M. Bradshaw, AAAI/MIT Press.

Wolpert, D. H., and Tumer, K. 2001. Optimal payoff functions for members of collectives. *Advances in Complex Systems* 4(2/3):265–279.

Wolpert, D. H.; Tumer, K.; and Frank, J. 1999. Using collective intelligence to route internet traffic. In *Advances in Neural Information Processing Systems - 11*, 952–958. MIT Press.

Wolpert, D. H.; Wheeler, K.; and Tumer, K. 2000. Collective intelligence for control of distributed dynamical systems. *Europhysics Letters* 49(6).

Wolpert, D. H. 2001a. Bounded-rationality game theory. pre-print.

Wolpert, D. H. 2001b. The mathematics of collective intelligence. pre-print.

Zhang, Y. C. 1998. Modeling market mechanism with evolutionary games. *Europhysics Letters*.